

LISTEN.
THINK.
SOLVE.®

INTEGRATED PRODUCTION & PERFORMANCE SUITE



Data Management

FactoryTalk® Historian SE



LIVE DATA INTERFACE USER GUIDE

PUBLICATION HSELD-UM023A-EN-E-May 2012

Supersedes Publication HSELD-UM022A-EN-E



Allen-Bradley • Rockwell Software

Rockwell
Automation

• • Copyright • •

Contact Rockwell Automation

Customer Support Telephone — 1.440.646.3434

Online Support — <http://www.rockwellautomation.com/support>

Copyright Notice

© 2012 Rockwell Automation Technologies, Inc. All rights reserved. Printed in USA.

© 2010 OSIsoft, Inc. All rights reserved.

This document and any accompanying Rockwell Software products are copyrighted by Rockwell Automation Technologies, Inc. Any reproduction and/or distribution without prior written consent from Rockwell Automation Technologies, Inc. is strictly prohibited. Please refer to the license agreement for details.

Trademark Notices

FactoryTalk, Rockwell Automation, Rockwell Software, the Rockwell Software logo are registered trademarks of Rockwell Automation, Inc.

The following logos and products are trademarks of Rockwell Automation, Inc.:

FactoryTalk Historian Site Edition (SE), FactoryTalk Historian Machine Edition (ME), RSView, FactoryTalk View, RSView Studio, FactoryTalk ViewStudio, RSView Machine Edition, RSView ME Station, RSLinx Enterprise, FactoryTalk Services Platform, FactoryTalk Live Data, and FactoryTalk VantagePoint.

The following logos and products are trademarks of OSIsoft, Inc.:

PI System, Sequencia, Sigmapine, gRecipe, sRecipe, and RLINK.

Other Trademarks

ActiveX, Microsoft, Microsoft Access, SQL Server, Visual Basic, Visual C++, Visual SourceSafe, Windows, Windows ME, Windows NT, Windows 2000, Windows Server 2003, and Windows XP are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Adobe, Acrobat, and Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

ControlNet is a registered trademark of ControlNet International.

DeviceNet is a trademark of the Open DeviceNet Vendor Association, Inc. (ODVA).

Ethernet is a registered trademark of Digital Equipment Corporation, Intel, and Xerox Corporation.

OLE for Process Control (OPC) is a registered trademark of the OPC Foundation.

Oracle, SQL*Net, and SQL*Plus are registered trademarks of Oracle Corporation.

All other trademarks are the property of their respective holders and are hereby acknowledged.

Restricted Rights Legend

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

Warranty

This product is warranted in accordance with the product license. The product's performance may be affected by system configuration, the application being performed, operator control, maintenance, and other related factors. Rockwell Automation is not responsible for these intervening factors. The instructions in this document do not cover all the details or variations in the equipment, procedure, or process described, nor do they provide directions for meeting every possible contingency during installation, operation, or maintenance.

This product's implementation may vary among users.

This document is current as of the time of release of the product; however, the accompanying software may have changed since the release. Rockwell Automation, Inc. reserves the right to change any information contained in this document or the software at anytime without prior notice. It is your responsibility to obtain the most current information available from Rockwell when installing or using this product.

| | |
|---|-----------|
| Introduction | 1 |
| Installation | 1 |
| Features | 2 |
| Related Documentation | 3 |
| Technical Support | 3 |
| Knowledgebase | 3 |
| Worldwide Support..... | 4 |
| Training Programs | 4 |
| Consulting Services | 4 |
| TechConnect Support..... | 4 |
| Principles of Operation..... | 7 |
| Processing Loops | 8 |
| FactoryTalk Diagnostics | 9 |
| Configuring FactoryTalk Historian Live Data Interface..... | 11 |
| Configuring the FTLD Interface Using the FactoryTalk Administration Console..... | 12 |
| Configuring the FTLD Interface Using the Interface Configuration Utility (ICU) | 12 |
| Adding Remote Servers to Connection Manager | 16 |
| Defining Remote Servers as API Hosts | 16 |
| Point Source | 17 |
| PI 3 Server Node: Reserved Point Sources | 17 |

| | |
|---|-----------|
| Point Configuration | 19 |
| Point Attributes | 19 |
| Length | 20 |
| Point Source | 20 |
| Point Type | 20 |
| PI 3 Server Nodes | 20 |
| Location1 | 21 |
| Location2 | 21 |
| Location3 | 21 |
| Location4 | 22 |
| Advise and Polled Tags | 22 |
| Output Tags | 22 |
| Location5 | 22 |
| InstrumentTag | 23 |
| Length | 23 |
| ExDesc | 23 |
| Length | 23 |
| SourceTag | 24 |
| Compression Testing | 24 |
| Exception Reporting | 25 |
| Output Points | 25 |
| Trigger Method 1 (Recommended) | 26 |
| Trigger Method 2 | 26 |
| Sample Tag Configurations | 27 |
| Scan Classes | 27 |
| Polled Tags | 27 |
| Advise Tags | 28 |
| Event Tags | 28 |
| I/ORates Tag Configuration | 31 |
| Monitoring I/ORates Tag on the Interface Node | 31 |
| Configuring I/ORates Tag with ICU | 31 |

| | |
|---|-----------|
| Performance Point Configuration | 35 |
| Startup Command File | 37 |
| Command-line Parameters | 37 |
| Sample FTLDInt.bat File | 41 |
| Setting File | 42 |
| Interface Node Clock | 45 |
| Security | 47 |
| Starting and Stopping the Interface..... | 49 |
| Using the Interface Configuration Utility | 49 |
| Using the Administrative Tools | 50 |
| Error and Informational Messages | 51 |
| System Errors and PI Errors..... | 52 |
| Error Descriptions | 53 |
| Index..... | 55 |

Introduction

The FactoryTalk Historian system uses the PI system as a back end to store tag data. The FactoryTalk Historian Live Data Interface (FTLD) allows FactoryTalk Historian to connect with Rockwell data sources. The FTLD interface provides buffering capability before the data is permanently stored in the Historian archive.

The FTLD Interface is integrated with the FactoryTalk Live Data service provided by FactoryTalk Services Platform. It is a FactoryTalk Live Data client that relies on FactoryTalk Live Data service to talk to FactoryTalk data servers such as FactoryTalk View SE, RSLinx Enterprise, and OPC servers that are part of a FactoryTalk application.



The FactoryTalk Historian documentation uses the terms tag and point to mean the same thing. An example of a tag name is:
FTLDEnt:rc:RSLinxC.secondArray000

The FTLD Interface is configured using the FactoryTalk Administration Console or Interface Configuration Utility. See *Configuring FactoryTalk Historian Live Data Interface* (page 11) for more information.

Installation

The steps to install the FactoryTalk Historian Live Data Interface are explained in the *FactoryTalk Historian SE Installation and Configuration Guide*, available from **Start > All Programs > Rockwell Software > FactoryTalk Historian SE > Help**.



The complete user documentation on FactoryTalk Historian SE is available in the **Common Files > Rockwell > Help** folder in your **Program Files** directory, and in the **Redist > Docs** folder on your FactoryTalk Historian SE installation DVD.

Features

The features of the FTLD Interface include:

| | |
|------------------------------|---|
| PI SDK | The PI SDK and the PI API are bundled together and must be installed on each PI Interface node. The FTLD Interface does not specifically make PI SDK calls. It uses the PI API calls to support longer instrument tag fields and multiple-character point sources. The PI SDK cannot be used if the interface will be set up to use Disconnected Startup because it is based on API calls only. |
| Source of Time Stamps | The FTLD Interface can accept time stamps from the FactoryTalk Live Data server or it can provide time stamps from the FactoryTalk Historian server. This is controlled by a command-line parameter. See <i>Startup Command File</i> (page 37) for more information on using the command line to control time stamps. |
| UniInt-based | UniInt stands for Universal Interface. UniInt is not a separate product or file; it is an OSIsoft-developed template used by developers and is integrated into many interfaces, including this interface. The purpose of UniInt is to keep a consistent feature set and behavior across as many of Rockwell interfaces as possible. It also allows for a very rapid development of new interfaces. In any UniInt-based interface, the interface uses some of the UniInt-supplied configuration parameters and some interface-specific parameters. |
| Platforms | The FTLD Interface is designed to run on the Microsoft Windows operating systems. Due to its dependency on FactoryTalk Services Platform, the FactoryTalk Historian Live Data Interface is not supported on non-Windows platforms. To see a list of operating systems supported, refer to the <i>FactoryTalk Historian SE Release Notes</i> . |

Related Documentation

The following documentation provides additional information related to the FTLD Interface:

- *FactoryTalk Historian SE Installation and Configuration Guide*,
 - *FactoryTalk Historian SE Release Notes*,
- both available from **Start > Programs > Rockwell Software > FactoryTalk Historian SE > Help**.
- *FactoryTalk Help*, available from **Start > Programs > Rockwell Software > FactoryTalk Tools > FactoryTalk Help**.



To access other documents on FactoryTalk Historian SE, go to the **Common Files > Rockwell > Help** folder in your **Program Files** directory.

Technical Support

Please visit *Rockwell Automation Customer Support Center* (<http://www.rockwellautomation.com/support/>) for access to user forums, sample code, software and firmware updates, product manuals, and other downloads.

Knowledgebase

The Customer Support Center offers an extensive online knowledgebase that includes frequently asked questions (FAQs) and the latest patches. Please visit the *support site* (<http://www.rockwellautomation.com/resources/support.html>) and select the **Knowledgebase** link located under **Tools & Resources** to:

- View technical and application notes.
- Obtain software patches and firmware updates.

- Subscribe to product and service e-mail notifications.
- Ask questions.

Worldwide Support

If you are not located in North America and want to contact Rockwell Automation Support, use the *Worldwide Locator* (<http://www.rockwellautomation.com/locations/>) for worldwide contact information.

Training Programs

Rockwell Automation offers a wide range of training programs that include e-learning, regularly scheduled and custom-tailored classes, self-paced training, and certificate programs. If you would like more information on training, visit the *Rockwell Automation Training site* (<http://www.rockwellautomation.com/services/training/>) or call 1.440.646.3434.

If you are not located in North America and want to contact Rockwell Automation Support, use the *Worldwide Locator* (<http://www.rockwellautomation.com/locations/>) for worldwide contact information.

Consulting Services

If you are not located in North America and want to contact Rockwell Automation Support, use the *Worldwide Locator* (<http://www.rockwellautomation.com/locations/>) for worldwide contact information.

TechConnect Support

With TechConnect Support, your site has unlimited, real-time access to Rockwell Automation's global network of Customer Support Centers and

technical resources. TechConnect service levels are provided at the *TechConnect site* (<http://www.rockwellautomation.com/services/onlinephone/techconnect/>).

When you contact Rockwell Technical Support, please provide:

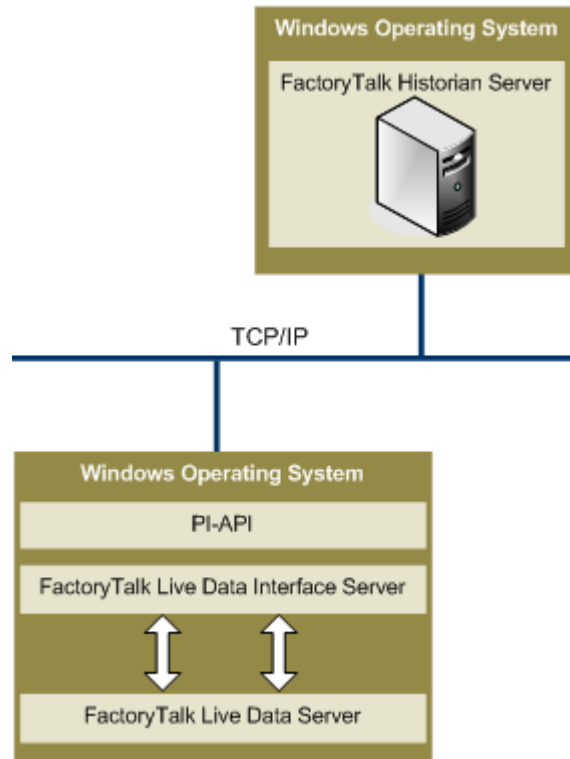
- Product name, version, and/or build numbers.
- Computer platform (CPU type, operating system, and version number).
- Exact wording of any messages that appeared on your screen.
- The message log(s) at that time.
- Descriptions of:
 - What happened and what you were doing when the problem occurred.
 - How you tried to solve the problem.

-
-
- FactoryTalk Historian Live Data Interface User Guide
-
-

Principles of Operation

The FactoryTalk Live Data (FTLD) Interface is a FactoryTalk Live Data client that enables process data to be passed between a FactoryTalk Live Data server (for example, RSLinx Enterprise) and a FactoryTalk Historian server. Each instance of the FTLD Interface can provide data to a single FactoryTalk Historian SE server or collective. Multiple instances of the interface may be configured, if required, if API buffering is used.

The figure below shows the basic workflow of the FactoryTalk Historian Live Data Interface.



Processing Loops

At startup, the FTLD Interface tries to establish connection to both the FactoryTalk Live Data server and the FactoryTalk Historian server.

Once startup is complete, the Interface enters the processing loop, which includes:

- Servicing scheduled input points. Each Scan Class is processed in turn.

- Servicing output points as events arrive.
- Servicing triggered input points as events arrive.

The Historian Point Database is checked every 2 minutes for points that are added, edited, and deleted. When point updates are detected, the points are loaded (or reloaded) by the Interface as appropriate. The 2-minute update interval can be adjusted with the */updateinterval* command-line parameter discussed in the *UniInt Interface User Manual*.

The manual is located in the *UniInt* folder (*c:\Program Files\Rockwell Software\FactoryTalk Historian\Server\PIPC\Interfaces\UniInt*).

The Interface processes a maximum of 25 point updates at a time. If more than 25 points are added, edited, or deleted at one time, the Interface will process the first 25 points, wait 30 seconds (or the length of time specified by the */updateinterval* parameter, whichever is lower), process the next 25 points, and so on. After all points have been processed, the Interface will resume checking for updates every 2 minutes (or the length time specified by the */updateinterval* parameter).

All tag edits are performed in the following way: old versions of edited tags are deleted from the interface, new versions are added in. Therefore, it is more efficient to stop and restart the interface if a large number of tags are edited.

FactoryTalk Diagnostics

The FTLD Interface sends messages about its operation to FactoryTalk Diagnostics.

FactoryTalk Diagnostics provides the following information about the FTLD Interface:

- Informational messages on the interface startup and shutdown.
- The scan rate of each scan class.

- The count of points loaded by the interface.
- Error messages for points rejected by the interface because they are configured incorrectly.
- Error messages for points rejected by the FTLD server or error messages sent from the FTLD server.

Because the FTLD Interface is based on the PI-UniInt framework, a few messages are sent to the PIPC log by the PI-UniInt. These error messages are produced by the standard OSIssoft interface routines or by the PI API.



The FactoryTalk Diagnostics Setup and Viewer are available from the **Tools** menu of FactoryTalk Administration Console.

To learn more about viewing error messages and accessing the log file, refer to the help files for FactoryTalk Diagnostics Setup and FactoryTalk Diagnostics Viewer.

Configuring FactoryTalk Historian Live Data Interface

To configure a FactoryTalk Historian Live Data Interface (FTLD), we recommend that you use FactoryTalk Administration Console.

However, you must use the Interface Configuration Utility (ICU) to configure an FTLD Interface in the following cases:

- If you are configuring redundant FTLD Interfaces.

For more information, refer to the 59932 article at the *Rockwell Automation Knowledgebase*

(https://rockwellautomation.custhelp.com/app/answers/detail/a_id/59932).

- If you are configuring the buffer subsystem on an FTLD Interface.

For more information, refer to "Enable Buffering" in the *FactoryTalk Historian SE Installation and Configuration Guide*.



Enabling the buffer subsystem is a recommended step during the installation of the FactoryTalk Historian Live Data Interface. To take advantage of this feature, you must install the FTLD Interface on a separate machine than the FactoryTalk Historian SE.

- If you are configuring the **Disconnected Startup** option.

For more information, refer to "(Unint) Disconnected Startup" in the *Interface Configuration Utility Help*.



To open the Interface Configuration Utility (ICU) online help, go to **Start > All Programs > Rockwell Software > FactoryTalk Historian SE > Interface Configuration Utility**, and select **Help > Contents and Index** from the main menu.



If you create an FTLD Interface using the Interface Configuration Utility, it will not appear in the list of interfaces in the FactoryTalk Administration Console. As a result, you will not be able to configure the interface from the FactoryTalk Administration Console.

Configuring the FTLD Interface Using the FactoryTalk Administration Console

Refer to "Configuring FactoryTalk Historian Live Data Interface" in the *FactoryTalk Historian SE Installation and Configuration Guide* for information on configuring an FTLD Interface using the FactoryTalk Administration Console. This section also includes steps on how to enable buffering on the remote interface.

You can configure the FTLD Interface on your local computer (the computer on which the FactoryTalk Historian SE server is installed); however, we recommend that you always install the FTLD Interface on the computer that has the data server installed. Refer to "Verifying the FactoryTalk Historian Live Data Local Interface" in the *FactoryTalk Historian SE Installation and Configuration Guide* for more information.

Configuring the FTLD Interface Using the Interface Configuration Utility (ICU)



For more information on the Interface Configuration Utility, refer to the *FT Historian SE Interface Configuration Utility* guide available in the **Common Files > Rockwell > Help** folder in your **Program Files** directory.

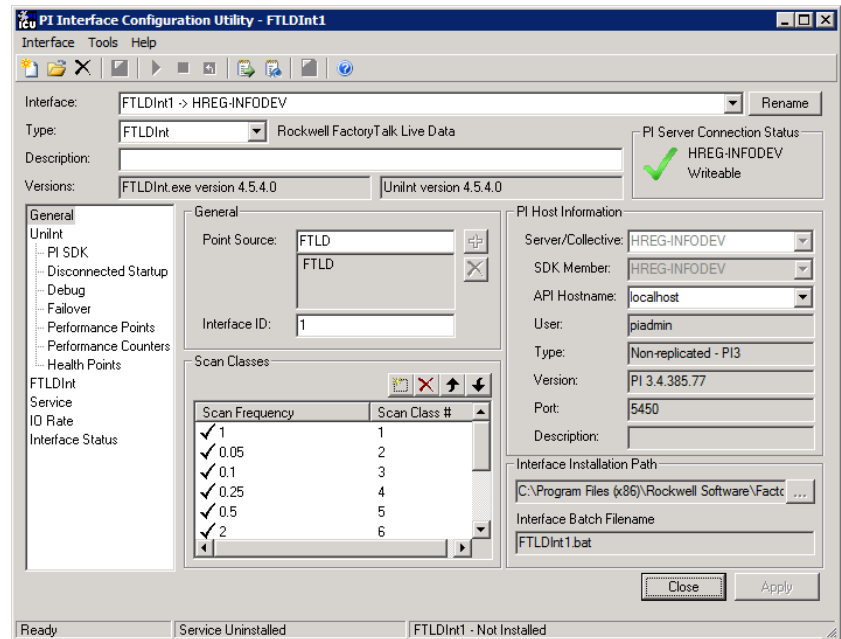


If you configure an interface in the ICU, the batch file of the interface (*FTLDInt.bat*) will be maintained by the ICU, and all configuration changes will be kept in that file.

To configure an FTL D Interface with the ICU:

1. Go to **Start > All Programs > Rockwell Software > FactoryTalk Historian SE > Interface Configuration Utility**. The **Interface Configuration Utility** dialog box appears.
2. From the menu bar, select **Interface > New Windows Interface Instance from BAT File**. The **Open Interface Configuration File** dialog box appears.
3. Navigate to *C:\Program Files\Rockwell Software\FactoryTalk Historian\Server\PIPC\Interfaces\LDInterface*.
4. Select the *FTLDInt_FTL D.bat.bak* file and click **Open**.
5. Click **OK** in the **Interface Configuration Utility** message box to acknowledge the information on duplicated point sources and ID combinations.

The FTLD Interface configuration is displayed on the **General** page of the ICU.



6. From the **Type** drop-down list, select **FTLDInt**, if it has not been chosen automatically.
7. If necessary, edit the **Scan Frequency** and **Scan Class** values.

| | |
|-----------------------|--|
| Scan Frequency | Indicates the frequency at which the interface reads values from the FactoryTalk Live Data server. |
| Scan Class # | Associated with the Scan Frequency value. If there is no interface defined, the Scan Class # will be displayed without a time period. |

8. Click **Apply**.

Note that in this example the **API Hostname** is *localhost*, which means that the interface will be configured to communicate with the local FactoryTalk Historian SE server.

If you want the interface to communicate with a remote FactoryTalk Historian server, you need to:

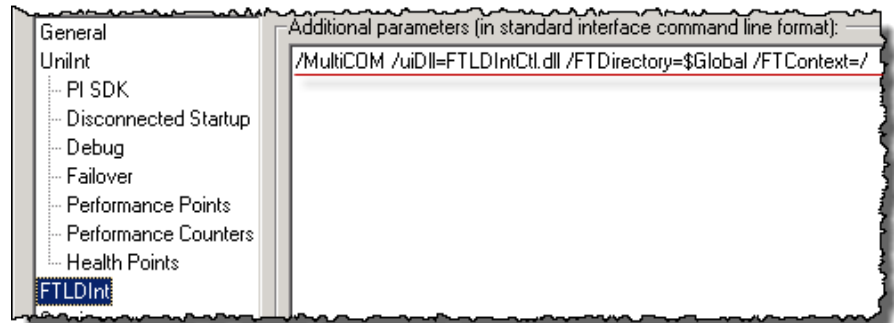
9. *Add the remote server to the Connection Manager (page 16).*
10. *Define the remote server as the API host (page 16).*

Once you have added the server to the Connection Manager and defined it as the API host, you can select the server from the **API Hostname** drop-down list on the **General** page in the ICU.

There are additional parameters available for configuring the interface:

- /MultiCOM
- /uiDll
- /FTDirectory
- /FTContext

To view them, click **FTLDInt** in the left pane of the dialog box.



See *Command-line Parameters* (page 37) for more information.

Adding Remote Servers to Connection Manager

To add a remote server to the Connection Manager:

1. In the ICU, select **Interface > SDK Connections**.
2. In the **Connection Manager** dialog box, select **Server > Add Server**. The **Add Server** dialog box appears.
3. In the **Network Node** text box, type the name of the remote server, and click **OK**.
4. In the **Connection Manager** dialog box, click **Save**.

Defining Remote Servers as API Hosts

To use a remote server as the API host:

1. In the ICU, select **Interface > SDK Connections**. The **Connection Manager** dialog box appears.
2. From the menu bar, select **Tools > Options**. The **Connection Options** dialog box appears.
3. From the **Default Server** list, select the remote server you want to use as the API host and click **OK**.
4. In the **Connection Manager** dialog box, click **Save**.

Point Source

The FactoryTalk Historian Live Data Interface uses FTLTD as the Point Source. A Point Source is a unique, single- or multi-character string that is used to identify a Historian point as a point that belongs to a particular interface.

For example, the string Boiler1 may be used to identify points that belong to the MyInt Interface. To implement this, the Point Source attribute would be set to Boiler1 for every Historian Point that is configured for the MyInt interface. Then, if */ps=Boiler1* is used on the start-up command line of the MyInt interface, the interface will search the Historian Point Database upon startup for every Historian point that is configured with a Point Source of Boiler1.

Before an interface loads a point, it also examines additional Historian point attributes to determine whether a particular point is valid for the interface. For additional information, see *Command-line Parameters* (page 37).

PI 3 Server Node: Reserved Point Sources

Several subsystems and applications that are shipped with PI System are associated with default Point Source characters:

| Subsystem | Point Source character |
|-----------|------------------------|
| Totalizer | T |
| Alarm | G and @ |
| Random | R |

| Subsystem | Point Source character |
|-----------------------|------------------------|
| RampSoak | 9 |
| Performance Equations | C |

Do not use these Point Source characters or change the default Point Source characters for these applications.

If a Point Source character is not explicitly defined when creating a PI point, the point is assigned a default Point Source character of *Lab* (PI 3). Therefore, do not use Lab as a Point Source character for an interface to avoid confusion.



Do not use a Point Source character that is already associated with another interface program. However, it is acceptable to use the same Point Source for multiple instances of an interface.

Point Configuration

A FactoryTalk Historian point is the basic building block for controlling data flow to and from the FactoryTalk Historian SE server. A single point is configured for each measurement value that needs to be archived.

Point Attributes

Historian points (tags) have approximately 50 attributes. These attributes define how data is to be collected and stored for the point. The proper configuration of these attributes is the key to optimizing the FactoryTalk Historian server for both data storage efficiency and quick retrieval. Each FactoryTalk Historian interface handles specific point attributes differently.

A tag is a label or name for a Historian point.



The FactoryTalk Historian documentation uses the terms tag and point to mean the same thing. An example of a tag name is:

FTLDEnt:rc:RSLinxC.secondArray000

The information presented in this chapter is necessary to define FactoryTalk Live Data (FTLD) points for use with a FactoryTalk Live Data server. Failing to correctly configure FactoryTalk Historian data points will result in poor or no communication between the interface and the FactoryTalk Live Data server. See *Error and Informational Messages* (page 51) for more information on errors that may occur.

Length

The Tag field allows a maximum of 1023 characters.

Point Source

The Point Source is a unique single or multiple character string that is used to identify the PI point as a point that belongs to a particular interface. The Point Source for FactoryTalk Historian Live Data Interface is *FTLD*. For additional information, see the */ps* command-line parameter description in *Command-line Parameters* (page 37).

Point Type

Typically, device point types do not need to correspond to Historian point types. For example, integer values from a device can be sent to floating-point or digital Historian tags. Similarly, a floating-point value from the device can be sent to integer or digital Historian tags, although the values will be truncated.

PI 3 Server Nodes

The following point types are supported on PI 3 Servers:

- float16
- float32
- float 64
- int16
- int32
- digital
- string



For more information on the individual point types, refer to the *FT Historian SE Server Reference Guide*, available in the **Common Files > Rockwell > Help** folder in your **Program Files** directory.

Location1

Location1 indicates to which copy of the interface the point belongs. The value of this attribute must match the */id* startup parameter. The default value for the FTLD Local Interface is *1*.

Location2

Location2 is not used for the FTLD Interface.

Location3

Location3 is used to define a data collection mode:

| | |
|-----------------------------|---|
| 0 - Polled or Event | Data is collected based on the scan rate and saved in the buffer. When the Historian server requests data, the data stored in the buffer is sent to the server. In this method, the values being sent to the Historian server are from a buffer, so they may not represent the exact (current) values in the controllers. |
| 1 - Advise (Default) | Data is collected only when a value changes in the controller. It is not based on the scan rate. It is the default data collection method and is the most efficient because data is sent to the Historian server only when the value changes. |
| 2 - Output | Data is written back to the Live Data server such as RSLinx Enterprise, HMI server such as FactoryTalk View, or OPC server such as RSLinx Classic. Use this method when you want to write data points back to the data servers. |

Location4

Location4 defines the scan class for the Historian point. The scan class determines the frequency at which input points are scanned for new values. For more information, see the description of the */f* parameter in *Command-line Parameters* (page 37).

The updates from the FactoryTalk Live Data server come in groups: at startup, the interface defines a group on the Live Data server and adds all points within the given scan class to the group. The FactoryTalk Live Data server is queried for all points within a group at the same time; therefore, some consideration should be given to the creation of scan classes. Having more than one scan class with the same scan period is allowed, and using different offsets on those scan classes may improve performance.

Advise and Polled Tags

Advise tags and Polled tags use Location4 to specify the requested update rate for the group.

Output Tags

Location4 is ignored for Output tags.



Advise, Polled, and Output data collection methods are explained in *Location3* (page 21).

Location5

If *Location5=1* and *Location3=0*, it will force an asynchronous read from the data server and should only be used for event-triggered points due to performance concerns.

InstrumentTag

InstrumentTag contains the ItemID of the tag. The format of this field depends on the FactoryTalk Live Data server:

| Type of the tag | InstrumentTag syntax |
|-----------------|---|
| Device tags | <application name>/<area name>:<data server name>:[<shortcut name>]<tag name> |
| HMI tags | <application name>/<area name>:<HMI server Name>:<folder name>\<tag name> |

The field must match the point defined on the FactoryTalk Live Data server, including punctuation, spaces, and case.

Length

InstrumentTag field allows a maximum of 1023 characters.

The FTL D interface gets the \$Global scope from the */FTDirectory* parameter and / from the */FTContext* parameter. For additional information on these parameters, see *Command-line Parameters* (page 37).

ExDesc

The ExDesc (Extended Descriptor) is a string attribute. Typically, this attribute is used to implement Trigger Input points.

For example: If a PI point has the ExDesc attribute *"EVENT='Tag1' Anychange,"* it means trigger on any change as long as the Tag1 value of the current event is different from the value of the previous event.

Length

The ExDesc field allows a maximum of 1023 characters.

SourceTag

An output point is associated with a trigger point by setting the SourceTag attribute of the output point equal to the tag name of the trigger point.

For more information, see *Output Points* (page 25).

Compression Testing

For each data point, you can set the following attributes to configure its compression testing specification:

| | |
|--|--|
| Compression Deviation (CompDev) | Specifies in engineering units how much a value may differ from the previous value before it is considered to be a significant value. As a rule of thumb, set CompDev to the precision of the data source or hardware (instrument). Set it a little "loose" to err on the side of collecting, rather than losing data. After collecting data for a while, go back and check the data for your most important tags and adjust CompDev, if necessary. Note: Setting the CompDev attribute value too low causes too little data compression and wastes space in the archive. Setting the value too high causes loss of useful data. For most flows, pressures, and levels, use a deviation specification of 1% or 2% of Span. For temperatures, the deviation should usually be 1 or 2 degrees. |
| Compression Minimum (CompMin) | A point is archived if the elapsed time since the previous time the point was saved is greater than or equal to the minimum time and the value has changed by more than the deviation. For data points associated with interfaces that send exception reports, set CompMin to 0. |
| Compression Maximum (CompMax) | A point is archived if the elapsed time since the previous time the point was saved is greater than the maximum time. The recommended maximum time specification is one work shift (for example, 8 hours). Duplicate values will be archived if the elapsed time exceeds CompMax. You typically set CompMax to the same value for all points in the system. |



For information on compression testing, refer to "Exception Reporting and Compression Testing" in the *FactoryTalk Historian SE Server Reference Guide*,

available in the **Common Files > Rockwell > Help** folder in your **Program Files** directory.

Exception Reporting

For each data point, you can set the following three attributes to configure its exception reporting specification:

| | |
|--------------------------------------|---|
| Exception Deviation (ExcDev): | Specifies in engineering units how much a point's value must change before the interface considers it as a significant value, and sends it to the server. As a general rule, you should set the exception slightly smaller than the precision of the instrument system. |
| Exception Minimum (ExcMin): | Specifies a limit on how frequently the interface can report values to the server. For example, if you want the interface to wait full ten minutes before reporting a new value to the server, then you would set the ExcMin attribute to ten minutes. ExcMin is typically set to zero. |
| Exception Maximum (ExcMax): | Specifies a limit on how long the interface can go without reporting a value to the Historian server. After the ExcMax time period, the interface sends the next new value to the server, regardless of whether the new value is different from the last reported value. |



For information on exception reporting, refer to "Exception Reporting and Compression Testing" in the *FactoryTalk Historian SE Server Reference Guide*, available from **Start > Programs > Rockwell Software > FactoryTalk Historian SE > Help**.

Output Points

Output points control the flow of data from the Historian Server to any destination that is external to the server, such as the FTL D server. The FTL D Interface uses *Location3=2* to indicate an output point.

Outputs are triggered for UniInt-based interfaces. That is, outputs are not scheduled to occur on a periodic basis. There are two mechanisms for triggering an output, as described in the sections that follow.

Trigger Method 1 (Recommended)

For Trigger Method 1, a separate trigger point must be configured. The output point must have the same point source as the interface. The trigger point can be associated with any point source, including the point source of the interface. Also, the point type of the trigger point does not need to be the same as the point type of the output point.

The output point is associated with the trigger point by setting the SourceTag attribute of the output point equal to the tag name of the trigger point. An output is triggered when a new value is sent to the Snapshot of the trigger point. The new value does not need to be different than the previous value that was sent to the Snapshot to trigger an output, but the time stamp of the new value must be more recent than the previous value.

If no error is indicated, then the value that was sent to the trigger point is also written to the output point. If the output is unsuccessful, then an appropriate digital state that is indicative of the failure is usually written to the output point. If an error is not indicated, the output still may not have succeeded because the interface may not be able to tell with certainty that an output has failed.

Trigger Method 2

For Trigger Method 2, a separate trigger point is not configured. To trigger an output, write a new value to the Snapshot of the output point itself. The new value does not need to be different than the previous value to trigger an output, but the time stamp of the new value must be more recent than the previous value.

Trigger Method 2 may be easier to configure than Trigger Method 1, but Trigger Method 2 has a significant disadvantage. If the output is unsuccessful, there is no tag to receive a digital state that is indicative of the failure, which is very important for troubleshooting.

Sample Tag Configurations

See the following sample tag configurations:

- *Scan Classes* (page 27)
- *Polled Tags* (page 27)
- *Advise Tags* (page 28)
- *Event Tags* (page 28)

Scan Classes

Scan classes are defined in the startup file. Each */F=* parameter defines a scan class, which is numbered in order.

For example, if the .bat file reads

/F=2 /F=1:00 /F=1:30:00 /F=00:00:05,00:00:01

then these scan classes have been defined:

- Scan Class 1 has a scan period of 2 seconds.
- Scan Class 2 has a scan period of 60 seconds.
- Scan Class 3 has a scan period of 5400 seconds (90 minutes).
- Scan Class 4 has a scan period of 5 seconds, with an offset of 1 second.

Polled Tags

Polled tags are read once every scan period. To set up a polled tag, set *Location1* to match the */ID* parameter, *Location3=0*, and *Location4=scanclass#*.

For example:

| Tag | InstrumentTag | Loc1 | Loc2 | Loc3 | Loc4 | Loc5 |
|------------|---------------|------|------|------|------|------|
| FiveSec.PV | ItemID1 | 1 | 0 | 0 | 1 | 0 |

| Tag | InstrumentTag | Loc1 | Loc2 | Loc3 | Loc4 | Loc5 |
|--------------|---------------|------|------|------|------|------|
| OneMin.PV | ItemID2 | 1 | 0 | 0 | 2 | 0 |
| NinetyMin.PV | ItemID3 | 1 | 0 | 0 | 3 | 0 |

Advise Tags

For Advise tags, the interface asks the FTLD server to send data when it changes, and how often it should read the device to see if there is a new value.

For example:

| Tag | InstrumentTag | Loc1 | Loc2 | Loc3 | Loc4 | Loc5 |
|------------------|---------------|------|------|------|------|------|
| AdvFiveSecs.PV | ItemID1 | 1 | 0 | 1 | 1 | 0 |
| AdvOneMin.PV | ItemID2 | 1 | 0 | 1 | 2 | 0 |
| AdvNinetyMins.PV | ItemID3 | 1 | 0 | 1 | 3 | 0 |

Event Tags

Event tags are read when the triggering event occurs. An event happens when the FactoryTalk Historian snapshot receives a value for the trigger tag. It may have the same time stamp and quality and value as the last event so the snapshot value for that trigger may seem the same, but the act of receiving a value for the trigger tag causes the interface to receive a notification that the trigger has been updated.

To configure triggered input tags, specify the name of the trigger tag in the ExDesc field using the following format:

`EVENT='triggertagname' event_condition`

where `triggertagname` is enclosed in single quotes and, if specified, the `event_condition` immediately follows the `triggertagname`.

If the `event_condition` is not specified then it defaults to `Anychange`.

The update rate for event items groups is also related with the scan class, so the server will be asked to update its cache once every scan period for every event tag defined. That is probably faster or slower than necessary. You must set the `Loc4` attribute to make event tags work well. The `Location 5` attribute should have the value 1 for Event tags.

Typical example:

| Tag | ExDesc | Instrument Tag | Loc1 | Loc2 | Loc3 | Loc4 | Loc5 |
|-------------|--------------------|----------------|------|------|------|------|------|
| PM1_Temp.PV | EVENT='PM1_Trigger | 'ItemID1 | 1 | 0 | 0 | 2 | 1 |
| PM1_Rate.PV | EVENT='PM1_Trigger | 'ItemID2 | 1 | 0 | 0 | 3 | 1 |

In this case, `PM1_Trigger` are tags that are updated by this interface, by another interface, or by manual entry. When `PM1_Trigger` gets a new event in the PI snapshot, the interface will send data for both `PM1_Temp.PV` and `PM1_Rate.PV` to the PI server.

-
-
- FactoryTalk Historian Live Data Interface User Guide
-
-

I/ORates Tag Configuration

An I/ORates tag measures the throughput of an FTLD Interface. In particular, the value of an I/ORate point represents a 10-minute average of the total number of values per minute that the FTLD Interface sends to the FactoryTalk Historian server. Because values are averaged over a 10-minute interval, the first calculated value is not written to the Historian server earlier than 10 minutes after the interface has started. You can configure one I/ORates tag for each copy of the FTLD Interface that is in use.



The Historian system documentation often uses the terms Event Counter Tag and I/ORate Point synonymously.

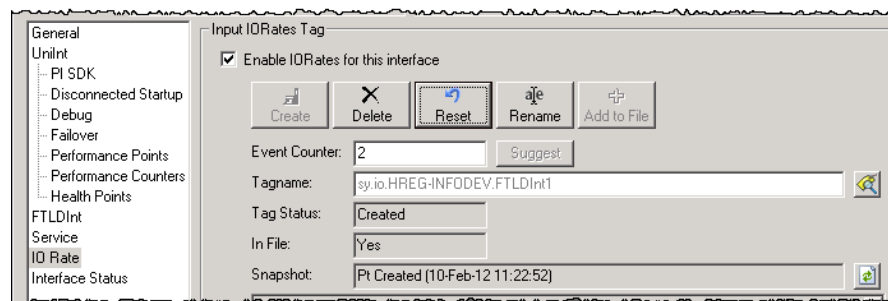
Monitoring I/ORates Tag on the Interface Node

For Windows nodes, the 10-minute rate averages (in events/minutes) can be monitored with a client application such as PI ProcessBook.

Configuring I/ORates Tag with ICU

The Interface Configuration Utility (ICU) provides a user interface for creating and managing the I/ORates tag.



To access the I/ORates tag data in ICU, select **IO Rate** from the left pane of the ICU dialog box.



ICU currently allows for one I/ORates tag to be configured for each copy of the interface that is in use. Some interfaces allow for multiple I/ORates tags.

The **Input IORates Tag** section contains the following elements:

| | |
|--|---|
| Enable IORates for this interface | Select the check box to enable I/ORates for the selected interface. Clear the check box to disable I/ORates for the selected interface. |
| Create | Click it to create the suggested I/ORates tag with the tag name indicated in the Tagname text box. |
| Delete | Click it to delete the I/ORates tag listed in the Tagname text box. |
| Reset | Click it to reset the I/ORates Event Counter and Tag settings. |
| Rename | Click it to change the name of the I/ORates tag using the Rename IORates Tag dialog box. |
| Add to File | Click it to add the tag to the <i>IORates.dat</i> file with the event counter listed in the Event Counter text box. |
| Event Counter | The Event Counter correlates a tag specified in the <i>iorates.dat</i> file with this copy of the interface. The command-line equivalent is <i>/ec=x</i> , where <i>x</i> is the same number that is assigned to a tag name in the <i>iorates.dat</i> file. |

| | |
|---|---|
| Suggest | <ul style="list-style-type: none"> ■ In the ICU dialog box, click it if you want the system to suggest the next available Event Counter. The button is active if the Event Counter text box is empty or contains an illegal value. ■ In the Rename IORates Tag dialog box, click it if you want the system to suggest a tag name. |
| Tagname | Type the name of the I/ORates tag, or click the TagSearch icon to find the tag using the Tag Search dialog box. |
|  | TagSearch. Click it to find the I/ORates tag for the interface using the Tag Search dialog box. |
| Tag Status | <p>Indicates whether the I/ORates tag exists in the Historian server. The text box may have the following values:</p> <ul style="list-style-type: none"> ■ Created The tag exists in the Historian server. ■ Not Created The tag does not yet exist in the Historian server. ■ Deleted The tag has just been deleted from the Historian server. ■ Unknown The ICU is not able to access the Historian server. |
| In File | Indicates whether the I/ORates tag listed in the tag name and the event counter is in the <i>IORates.dat</i> file. It may have the Yes or No values. |
| Snapshot | Holds the snapshot value of the I/ORates tag, if the I/ORates tag exists in the FactoryTalk Historian server. The value of the text box is updated when you click IO Rate in the left pane of the ICU dialog box, and when the interface is first started. You can refresh it manually by clicking the Refresh snapshot icon. |
|  | Refresh snapshot. Click it to refresh the snapshot. |

-
-
- FactoryTalk Historian Live Data Interface User Guide
-
-

Performance Point Configuration

Performance Point tags document how long it takes to complete a scan. Due to the architecture of this interface, the performance point tags are not valid - the server's response is asynchronous, so the time to scan bears no relation to the amount of time it may take to get the data from the server.

-
-
- FactoryTalk Historian Live Data Interface User Guide
-
-

Startup Command File

In Windows, command file names have a *.bat* extension. The Windows continuation character (^) allows for the use of multiple lines for the startup command. The maximum length of each line is 1024 characters (1 kilobyte). The number of parameters is unlimited, and the maximum length of each parameter is 1024 characters.

Command-line parameters should begin with a “/” character. For example, */ps=M*.

The Interface Configuration Utility (ICU) provides a tool for configuring the FTLD Interface startup command file.

Command-line Parameters

The table below lists command-line parameters and their descriptions.



We recommend that you always use the Interface Configuration Utility to modify the startup file. If you manually change the startup file and then open it using the Interface Configuration Utility, the utility will rewrite all the startup parameters in the file.

Required parameters:

| Parameter | Description |
|--|--|
| /ps=Source | <p>The /ps parameter specifies the point source for the interface. The Source value is not case sensitive. The length of the Source value is limited to 100 characters by PI-UniInt. The value can contain any character except '*' and '?'.</p> <p>The point source that is assigned with the /ps parameter corresponds to the Point Source attribute of the individual Historian point. The interface will attempt to load only those Historian points that have the appropriate point source.</p> <p>Strategies for assigning a point source character vary depending on the interpretation of the /id parameter by a particular interface. See the /id parameter description for more information.</p> |
| /id=x | <p>Example: /id=1</p> <p>The /id parameter specifies the interface identifier.</p> <p>The interface identifier is a string that is no longer than 9 characters in length. FTLD Interfaces also use the /id parameter to identify a particular interface copy number that corresponds to an integer value that is assigned to one of the Location code point attributes, most frequently Location1. For these interfaces, you should use only numeric characters in the identifier.</p> |
| /host=host:port | <p>The /host parameter specifies the PI Home node. Host is the IP address of the PI Server node or the domain name of the PI Server node. Port is the port number for TCP/IP communication.</p> <p>The port is always 5450.</p> |
| /uiDLL=FTLDIntCtl.dll /uiDLL="c:\Program Files\PIPC\Interfaces\PIUniInt\FTLDIntCtl.dll" | <p>The /uiDLL parameter is used to specify the DLL file name of the FTLD interface. The DLL file is usually installed in the Windows\System32 or Windows\SysWOW64 directory. If the DLL file is located elsewhere, the full path to the DLL must be provided in the /uiDLL parameter.</p> |
| /FTDirectory=\$Global | <p>The /FTDirectory parameter specifies the FactoryTalk directory. It can only be set to \$Global.</p> |
| /FTContext="/" | <p>This parameter is reserved. Set to "/".</p> |

| Parameter | Description |
|--|---|
| /f=SS or /f=SS,SS or /f=HH:MM:SS or /f=HH:MM:SS,hh:mm:ss | <p>Required for reading scanbased inputs.</p> <p>The /f parameter defines the time period between scans in terms of hours (HH), minutes (MM), and seconds (SS). The scans can be scheduled to occur at discrete moments in time with an optional time offset specified in terms of hours (hh), minutes (mm), and seconds (ss). If HH and MM are omitted, then the time period that is specified is assumed to be in seconds.</p> <p>Each instance of the /f parameter on the command-line defines a scan class for the interface. The first occurrence of the /f parameter on the command line defines the first scan class of the interface; the second occurrence defines the second scan class, and so on. Historian points are associated with a particular scan class via the Location4 point attribute. For example, all points that have Location4 set to 1 will receive input values at the frequency defined by the first scan class.</p> <p>Similarly, all points that have Location4 set to 2 will receive input values at the frequency specified by the second scan class, and so on.</p> <p>Two scan classes are defined in the following example: /f=00:01:00,00:00:05 /f=00:00:07 or, equivalently: /f=60,5 /f=7</p> <p>The first scan class has a scanning frequency of 1 minute with an offset of 5 seconds, and the second scan class has a scanning frequency of 7 seconds with no offset.</p> <p>When no offset is specified, the scan class will be scheduled for immediate execution. That is, the interface will not wait for a well-defined moment in time before scanning when no offset is specified.</p> <p>One can also specify sub-second scan classes on the command line such as: /f=0.5 /f=0.1</p> |

Optional parameters:

| Parameter | Description |
|-----------|---|
| /PISDK=# | <p>Default = 0.</p> <p>The /pisdsk parameter can be used to enable or disable the PI SDK.</p> <ul style="list-style-type: none"> ■ Use /pisdsk=1 to enable the PI SDK. ■ Use /pisdsk=0 to disable the PI SDK. |

| Parameter | Description |
|-----------------------|--|
| /maxstoptime=stoptime | <p>Default = 120 seconds.</p> <p>When an interface receives a signal from the operating system to shut down, it must perform a number of cleanup functions. If for some reason the execution of these functions takes longer time than the stop time, the interface will shut down without finishing these functions.</p> |
| /sio | <p>Default = send initial outputs.</p> <p>The /sio parameter stands for suppress initial outputs. The parameter applies only to interfaces, such as the FTLD, that support outputs. If the /sio parameter is not specified, the FTLD Interface will behave in the following manner:</p> <p>When the FTLD Interface is started, it determines the current Snapshot value of each output tag. Next, the FTLD interface writes this value to each output tag. In addition, whenever an individual output tag is edited while the FTLD Interface is running, it will write the current Snapshot value to the edited output tag.</p> <p>This behavior is suppressed if the /sio parameter is specified on the command line. That is, outputs will not be written when the FTLD Interface starts or when an output tag is edited. In other words, when the /sio parameter is specified, outputs will only be written when they are explicitly triggered.</p> |
| /perf=interval | <p>Default = 8 hours</p> <p>When the percentage of scans that a UniInt-based interface performs on time drops below 95%, UniInt will write the performance summaries for each scan class into the PIPC.log file.</p> <p>For example, if /perf=0.025, UniInt will write performance summaries every 90 seconds if the percentage of on-time scans is below 95%. The minimum time between summaries is 60 seconds. Setting /perf=0 disables summaries.</p> <p>If the /perf parameter is omitted, then by default, every 8 hours, UniInt checks whether summaries are needed.</p> <p>If the inputs for the interface are unsolicited, then performance summaries should be disabled by setting /perf=0 because performance summaries are meaningless for unsolicited input points.</p> |

| Parameter | Description |
|-------------|--|
| /q | <p>Default = no queuing.</p> <p>When the /q parameter is present, Snapshots and exceptions are queued before they are sent to the PI Server node. The maximum queue size is close to 4000 bytes. The queue is flushed between scans if it is not filled. For an interface collecting unsolicited data, the queue is flushed four times a second if it is not filled.</p> |
| /TSofPIorFT | <p>The /TSofPIorFT parameter specifies which time stamp will be used to determine when to send data to the Historian server:</p> <ul style="list-style-type: none"> ■ /TSofPIorFT=0 means using server time stamp ■ /TSofPIorFT=1 means using FTLD server time stamp <p>Notes:</p> <ul style="list-style-type: none"> ■ By default, there is no /TSofPIorFT parameter in the .bat file. In this case, the Historian Server time stamp will be used. ■ An unsolicited point will always use FTLD Server time stamps. |
| /MultiCom | <p>Use the /MultiCom parameter if you want the PI-UniInt executable to select the Multithreaded Threading model (COINIT_MULTITHREADED) for the call to CoInitializeEx(). Otherwise, the PI-UniInt will use the Apartment Threaded model (COINIT_APARTMENTTHREADED).</p> |

Sample FTLDInt.bat File

The following is a sample startup command file that comes with the installation:

```
"FTLDInt.exe"1 /uiDll=FTLDIntCtl.dll
/FTDirectory=$Global / FTContext=/ /PS=FTLD /ID=1
/host=localhost:5450 /ec=2 /pisdK=0 / maxstoptime=120
/q /sio /perf=8 /f=1 /f=0.05 /f=0.1 /f=0.25 / f=0.5
/f=2 /f=5 /f=10 /f=60 /f=120
```

Setting File

For more flexibility, the FTLD interface can use an INI format file to configure interface information. This format file does not install with the FTLD interface, but you can create the file manually. It should be named *FTLDInt.ini* and it must be created in the same path as the *FTLDInt.exe* file and the *FTLDIntCtl.dll* file.

The following is a sample setting file.

```
[FTLDIntSetting]
OnceMaxUnsolEvents=4
ScanClassToUpdateRate=1
PIOrFTLDTimestamp=0
FTLDResponseInterval=5000
```

Review the following for additional information:

| | |
|------------------------------|--|
| OnceMaxUnsolEvents | The OnceMaxUnsolEvents key specifies the maximum number of unsolicited events that the FTLD Interface can process in each loop for each advised tag. The default value for OnceMaxUnsolEvents key is 4. That is, the interface can process 4 events for each advised tag when the interface collects data. |
| ScanClassToUpdateRate | For polled points, the FTLD interface calculates the update rate from its scan time. The update rate is used while adding an item to the FTLD server. The algorithm is "Update Rate = Scan Time / ScanClassToUpdateRate" For example, if a point has a 1-second scan period and the ScanClassToUpdateRate is 2, the FTLD Interface will use 500 ms as the update rate. |
| PIOrFTLDTimestamp | The PIOrFTLDTimestamp key specifies which time stamp will be used to determine when to send data to the FactoryTalk Historian Server. PIOrFTLDTimestamp=0 means the FactoryTalk Historian server time stamp is used, and 1 means the FTLD server time stamp is used. This setting is similar to the /TSofPIorFT command-line parameter. See <i>Command-line Parameters</i> (page 37) for more information. |

| | |
|-----------------------------|--|
| FTLDResponseInterval | The FTLDResponseInterval key specifies the time interval during which the FTLD interface checks for data points that were marked 'bad_quality' by the FTLD service. The FTLD service will mark a data point 'bad_quality' if the data point links to an incorrect item (data point) in the FactoryTalk Diagnostics or if the FTLD service was not able to receive a data point from the FTLD interface because several others data points were being added to the FactoryTalk Historian server at that time. The period can be set in milliseconds (ms). The default is 5000 ms. |
|-----------------------------|--|

-
-
- FactoryTalk Historian Live Data Interface User Guide
-
-

Interface Node Clock

Make sure that the time and time zone settings on the computer are correct. Check the settings in the **Date and Time** program of Control Panel. If the locale where the interface node resides observes Daylight Saving Time, check **Automatically adjust clock for Daylight Saving Time** in **Time Zone Settings**.

Make sure that the TZ environment variable is not defined on the computer. To check it, type `set` in the Command Prompt window. All the environment variables currently defined in Windows will be listed. If the TZ environment variable is defined, remove it using the **System** program of Control Panel.



It is possible for computer nodes to boot up with different clock times before synchronizing to the time server. If synchronization is left to the Microsoft defaults, it can take several minutes for the system to synchronize all node clocks in the network. To minimize clock synchronization time, you can modify the registry on the computer nodes in the FactoryTalk Historian SE system to adjust local clocks to the time server. To do this, set the *MaxAllowedPhaseOffset* entry to *1* on every node in the FactoryTalk Historian SE system. The entry resides in the following subkey:

*My Computer\HKEY-
_LOCAL_MACHINE\System\ControlSet001\services\W32Time\Config.*



We recommend that only advanced users modify the registry. Refer to the system documentation for more information on the Windows registry.

-
-
- FactoryTalk Historian Live Data Interface User Guide
-
-

Security

The Trust Database must be configured so that the FTLD Interface is allowed to write data to the FactoryTalk Historian server.

If the FTLD Interface cannot write data to the FactoryTalk Historian server because it has insufficient privileges, a *10401* error will be reported in the *pipc.log* file. The file is located in *C:\Program Files\Rockwell Software\FactoryTalk Historian\Server\PIPC\DAT*.



For more details on trust configuration, please refer to the FactoryTalk Historian SE server manuals, available in the **Common Files > Rockwell > Help** folder in your **Program Files** directory.

-
-
- FactoryTalk Historian Live Data Interface User Guide
-
-

Starting and Stopping the Interface

Once you have installed the FTLD interface as a service, you can start and stop it in two ways:

- *Using the Interface Configuration Utility (ICU) (page 49).*
- *Using the Administrative Tools program of Control Panel (page 50).*


The FTLD interface service may terminate immediately after the startup for a variety of reasons. One of typical reasons is that the service is not able to find the command-line parameters in the associated *.bat* file. To avoid it, make sure that the root name of the *.bat* file and the *.exe* file are the same, and that the files are located in the same directory. Usually, they are stored in *C:\Program Files\Rockwell Software\FactoryTalk Historian\Server\PIPC\Interfaces\LDInterface*.

Further troubleshooting of the service may require consulting the *pipc.log* file, Windows Event Viewer, or other sources of log messages. See *Error and Informational Messages* (page 51) for additional information.


Using the Interface Configuration Utility

To start the FTLD interface:

1. Click **Start > All Program > Rockwell Software > FactoryTalk Historian SE > Interface Configuration Utility**. The **Interface Configuration Utility** dialog box appears.
2. From the **Interface** drop-down list, select the interface you want to start.

3. On the tool bar, click .
4. Wait, until the status on the status bar at the bottom of the dialog box changes to *Running*.

To stop the FTLD interface:

In the ICU, click  on the tool bar. The service status on the status bar changes to *Stopped*.

You may additionally check the status of the service in the **Administrative Tools** (page 50) program of Control Panel.

Using the Administrative Tools

To start the FTLD interface:

1. Click **Start > All Programs > Administrative Tools > Services**. The **Services** dialog box appears.
2. Right-click **FTLD<X>** (where *X* is the interface number) and select **Start**.
3. Wait until the status of the service changes to *Started*.

To stop the FTLD interface:

In the **Services** dialog box, right-click **FTLD<X>** and select **Stop**.

Error and Informational Messages

The FTLD Interface is based on the PI-UniInt framework, therefore a few error messages are sent to the PIPC log by PI-UniInt. Other error messages are sent to the FactoryTalk Diagnostics system. When troubleshooting, we recommend that you check both FactoryTalk Diagnostics and the PIPC log.

The following is the list of error messages sent to FactoryTalk Diagnostics.

| Severity | Message text |
|----------|---|
| Error | Failed to disconnect from FactoryTalk. |
| Error | Failed to initialize COM library. |
| Error | Failed to initialize COM security. |
| Error | Missing or invalid interface ID parameter. |
| Error | Failed to initialize FactoryTalk Diagnostics. |
| Error | Failed to launch FTLD interface due to the lack of the parameter /FTDirectory. |
| Error | Failed to launch FTLD interface due to the lack of the parameter /FTContext. |
| Error | Failed to connect to FactoryTalk Directory scope %s. |
| Error | System error: Insufficient memory. |
| Warning | Failed to remove FactoryTalk Live Data item <%s>. |
| Warning | PI point <%s> is refused because of failure to add FactoryTalk Live Data item <%s>. |
| Warning | PI point <%s> is refused because of the invalid attribute of Location [3]. |

| Severity | Message text |
|-------------|--|
| Warning | PI point <%s> is refused because of the invalid attribute of InstrumentTag. |
| Warning | PI point <%s> is refused because of the invalid attribute of Location [4]. |
| Warning | The value quality of PI point <%s (PointID: %d)> with FactoryTalk Live Data item <%s> is bad. |
| Warning | Failed to convert PIEvent to FTLD variant type of data. |
| Warning | Failed to write value to FactoryTalk Live Data item <%s>. |
| Warning | Monitor disconnected from FactoryTalk service. |
| Warning | Monitor reconnected to FactoryTalk service successfully. |
| Warning | The value quality of PI point <%s (PointID: %d)> with FactoryTalk Live Data item <%s> is bad because the FTLD service doesn't response in %d ms. |
| Information | PI point <%s> has been removed from the FTLD interface (InterfaceID: %d). |
| Information | PI point <%s> is edited in the FTLD interface (InterfaceID: %d). |
| Information | PI point <%s> has been added to the FTLD interface (InterfaceID: %d). |
| Information | Connected to FactoryTalk Directory scope %s successfully. |
| Information | FTLD interface(ID:%d) has Scan Class %d = %s. |
| Information | Disconnected from FactoryTalk Directory %s successfully. |

System Errors and PI Errors

System errors are associated with positive error numbers. Errors related to PI are associated with negative error numbers.

Error Descriptions

The descriptions of system and PI errors can be obtained with the *pidiag* utility. It is a command-line utility (*pidiag.exe*) located in the *C:\Program Files\Rockwell Software\FactoryTalk Historian\Server\PI\Adm* directory.

The following is the syntax to retrieve error descriptions of the messages:

```
\PI\adm\pidiag -e error_num
```

-
-
- FactoryTalk Historian Live Data Interface User Guide
-
-

A

Adding Remote Servers to Connection Manager • 16
Advise and Polled Tags • 22
Advise Tags • 28

C

Command-line Parameters • 37
Compression Testing • 24
Configuring FactoryTalk Historian Live Data Interface • 11
Configuring I/ORates Tag with ICU • 31
Configuring the FTLD Interface Using the FactoryTalk Administration Console • 12
Configuring the FTLD Interface Using the Interface Configuration Utility (ICU) • 12
Consulting Services • 4

D

Defining Remote Servers as API Hosts • 16

E

Error and Informational Messages • 51
Error Descriptions • 53
Event Tags • 28
Exception Reporting • 25
ExDesc • 23

F

FactoryTalk Diagnostics • 9

Features • 2

I

I/ORates Tag Configuration • 31
Installation • 1
InstrumentTag • 23
Interface Node Clock • 45
Introduction • 1

K

Knowledgebase • 3

L

Length • 20, 23
Location1 • 21
Location2 • 21
Location3 • 21
Location4 • 22
Location5 • 22

M

Monitoring I/ORates Tag on the Interface Node • 31

O

Output Points • 25
Output Tags • 22

P

Performance Point Configuration • 35

-
- FactoryTalk Historian Live Data Interface User Guide
-
-

PI 3 Server Node

- Reserved Point Sources • 17
- PI 3 Server Nodes • 20
- Point Attributes • 19
- Point Configuration • 19
- Point Source • 17, 20
- Point Type • 20
- Polled Tags • 27
- Principles of Operation • 7
- Processing Loops • 8

R

- Related Documentation • 3

S

- Sample FTLInt.bat File • 41
- Sample Tag Configurations • 27
- Scan Classes • 27
- Security • 47
- Setting File • 42
- SourceTag • 24
- Starting and Stopping the Interface • 49
- Startup Command File • 37
- System Errors and PI Errors • 52

T

- TechConnect Support • 4
- Technical Support • 3
- Training Programs • 4
- Trigger Method 1 (Recommended) • 26
- Trigger Method 2 • 26

U

- Using the Administrative Tools • 50
- Using the Interface Configuration Utility • 49

W

- Worldwide Support • 4